2020/9/22

## 1、初识小程序

### App.instance:

**app.json**声明式配置文件

配置内容：指定项目信息

微信官方文档：全局配置：

[https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html#%E5%85%A8%E5%B1%80%E9%85%8D%E7%BD%AE](https://developers.weixin.qq.com/miniprogram/dev/reference/configuration/app.html" \l "%E5%85%A8%E5%B1%80%E9%85%8D%E7%BD%AE)

entryPagePath 小程序默认启动页

pages 页面路径页表 string[]

写的时候不需要写拓展名

window 全局默认窗口表现，导航条背景文本颜色等、窗口设置等

tabBar 底部tab栏的表现

最重要的属性：list，添加2-5个

networkTimeout 网络超时事件

debug 是否开启debug调试

**app.js**逻辑

**app.wxss**样式

定义全局共享样式（公用样式container

### pages/page1

**page1.json**（可选）页面配置文件

设置当前页面工作时的window配置，此处会覆盖app.json中的window设置（优先级高），只可以设置window中设置的属性

**page1.js** 页面逻辑文件

用于创建页面对象，处理页面生命周期控制和数据处理等完成页面的逻辑包括功能的实现

**page1.wxml**（weixin markup language）页面结构文件

用于定义页面中的元素结构，语法遵循XML语法（不允许使用单标签），定义页面结构，包括页面元素之间的关系、数据绑定等

**page1.wxss**（可选）（weixin style sheet）页面样式文件

用于定义页面样式，语法完全遵循css语法，拓展了css基本用法和长度单位（单位比css高级一点，主要是rpx响应式像素）

约定大于配置：相同的名字来找到对应文件

## 微信小程序逻辑和界面分离的结构

### 界面层：

页面结构wxml

页面样式wxss

展示逻辑层的数据

#### 界面层的数据绑定

数据绑定：WXML中的动态数据均来自于对应Page的data

简单绑定：数据绑定使用Mustache语法（双{}）将变量包起来

框架: 最大的特点: 让开发者必须按照特定的方式来编写代码, 然后它再按照特定形式来执行代码

明确:

1.数据在哪? 在当前页面数据的data属性中

2.绑定到哪里去? 希望绑定到哪里就在哪里使用(mustache语法)

data: 为页面提供数据（data就是界面和逻辑之间的桥梁）

mustache语法：

不能用在标签名和属性名上,只能用在**值**的位置上

可以使用在:

  1.innerHTML(类似)

  2.元素属性上

可以直接使用字面量和简单的逻辑运算符

    <text>{{100>50?'小宝贝':'小笨蛋'}}</text>

    <text>{{','}}</text>

    <text>今天中午吃什么\n</text>

特殊用途:当语法解析误解了true和false的情况可以用mustache来解决

    <checkbox checked="false">烤盘饭\n</checkbox>

    <checkbox checked="true">重庆小面\n</checkbox>

    <checkbox checked='{{false}}'>鸡排饭</checkbox>

#### 界面层的列表渲染

列表数据渲染

<!-- <view>

  <checkbox>JavaScript</checkbox>

  <checkbox checked="{{true}}">HTML</checkbox>

  <checkbox>CSS</checkbox>

</view> -->

<view wx:for="{{todoLists}}">

基本的循环用法 wx:for

1.明确页面结构中的循环体

2.删除多余的重复内容,只保留一个

3.在剩下的上面加上wx:for属性,属性值等于要遍历的数据源(数据源一定是一个对象)

4.在这个标签(循环体)内部使用item来代表当前被遍历的元素

ps:

给被遍历的对象定义名称:wx:for-item 及  wx:for-index

如果有全局属性名与item冲突,若想要从内部去拿全局的属性,则需要改变item的名字比如:

<view wx:for="{{todoLists}}" wx:for-item="aaa">

 之后在for里的item就用aaa来代替表示

<!-- wx:if 用来根据一个布尔值来决定显示或隐藏 -->

<text>{{index}}</text>

  <checkbox checked="{{item.compeleted}}">{{item.name}}</checkbox>

</view>

#### 界面层的事件处理

基本的事件使用：通过给组件添加bind+事件名的属性，属性的值指向一个定义在当前页面对象中的js方法

使用bindtap来点击触发事件

<view>

    <button bindtap="buttonTapHandle">开始点餐</button>

  </view>

#### 界面层的事件冒泡

事件冒泡：

<view bindtap="outsideHandle" style="width:100px;height:100px;background-color:lightpink;">

    <view bindtap="insideHandle" style="background-color:lightblue;height:50px;width:50px;"></view>

</view>

点击内层块的时候先输出内层块事件再输出外层块事件。此时为事件冒泡

怎样阻止冒泡：把bindtap改成catchtap（catch+事件名：绑定事件的同时阻止事件冒泡）

<view **bindtap**="outsideHandle" style="width:100px;height:100px;background-color:lightpink;">

    <view **catchtap**="insideHandle" style="background-color:lightblue;height:50px;width:50px;"></view>

</view>

#### 界面层的事件传参

参数不能直接传递，只能通过dataset来给事件处理函数指定参数

// wxml

<button bindtap="tap2Handle" data-name="zhangsan">点击</button>

// Js：

tap2Handle: function (e) {

//e.target拿到的就是点击的元素

//dataset指定是元素上所有以data-开头的属性集合

    console.log(e.target.dataset);

  },

Ps：事件处理函数中的this指向的还是页面对象，与html不一样

#### 事件相关总结

1、事件的基本使用：通过bind+事件，catch+事件 来绑定事件

2、小程序中事件冒泡和html中处理起来不一样，是使用catch+事件来处理的

3、如果需要给事件处理函数指定参数，只能通过dataset

### 逻辑层：

业务逻辑的实现。由JavaScript完成业务数据供给、界面事件处理

1、小程序并不是运行在浏览器中，所以没有BOM和DOM对象

// console.log(window); //undefined

// console.log(document); //undefined

2、小程序的js中有一些额外的成员（成员是全局的，在任何一个地方都可以调用）

（1）全局必须要有一个app方法：用于定义应用程序实例对象

（2）page1.js中要有page方法：用于定义页面对象

（3）getApp方法：用于获取全局应用程序实例对象

（4）getCurrentPages方法：用来获取当前页面的调用栈(数组最后一共就是当前页面)

（5）wx对象：用来提供核心API的

3、小程序是JS是支持CommonJS规范的

不支持exports.xxx导出方式，只支持module.exports导出方式